Docket No.: 1080.1045CIPD/RAG

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Fumio NAGASHIMA

Serial No.: (Div. of 08/919,254) Prior Examiner: A. Khatri

Filed: (Concurrently) Prior Group Art Unit: 2762

For: METHOD AND APPARATUS FOR OBJECT ORIENTED PROGRAMMING IN COMPONENT BUILDING, ITS STORAGE MEDIUM, USES, SUPPORTS AND OBJECT BETWEEN-NETWORK-DISPLAY

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Prior to the examination on the merits of the above application, please amend the application as follows.

IN THE SPECIFICATION:

Please **AMEND** the specification as follows:

Before the first line insert the sentence:

--This application is a divisional of application number 08/919,254, filed August 28, 1997, now allowed; which is a CIP application of serial number 08/855,986, filed May 14, 1997, now abandoned.--

Page 1, line 10, delete “an”;
line 21, change "objects a" to --objects,--;
line 22, delete "a"; delete "an";
Page 2, line 1, after "objects" (first occurrence) insert --,--;

line 2, delete "the";

Page 3, line 4, delete "an";

line 9, delete "the" (last occurrence);

line 10, after "object" insert --,--;

line 17, change "the" to --this concept of--; delete "taking in";

line 18, delete "such a concept"; delete "an";

line 19, delete "software"; change "complicated, a method of making" to --complicated software, the creation--;

line 20, change "up" to --of--; delete "a";

line 21, delete "the";

line 24, after "of" insert --the--.

Page 4, line 1, after "of" insert --the--;

line 3, change "of the methods is" to --was--;

line 5, delete "the" (second occurrence);

line 6, delete "a";

line 10, change "the" to --an--;

line 18, change "is" to --was--;

line 21, change "is" to --becomes--;

line 22, change "They are" to --This technique is--; change "above-mentioned" to --desire for--;

line 23, change "reuse" to --reusable objects; after "thus" insert --,--;

line 24, delete "the" (second occurrence).

Page 5, line 3, change "number" to --amount--;

line 4, change "softwares" to --software--; change "are" to --is--; delete "a";

line 5, change "concept referred to an object-oriented." to --an object-oriented concept.--;

line 8, change "soft ware" to --software--.

Page 53, line 14, change "processings for an" to --processing for--;

Page 54, line 1, change "an" to --the--;

line 4, change "an" to --the--;

line 7, change "an" to --the--;

line 10, change "an" to --the--;

line 14, change "an" to --the--;

line 16, change "an" to --the--.

Page 55, line 9, change "processings for an" to --processing for--;

line 12, delete "of processings";

line 15, delete "of processings";

line 18, delete "of processings";

line 21, delete "of processings";

line 24, delete "of processings".

Page 56, line 3, delete "of processings";

line 6, change "processings for an" to --processing for--;

line 10, delete "of processings";

line 13, delete "of processings";

line 16, delete "of processings";

line 19, delete "of processings";

Page 57, line 14, change "(B)" to --38(B)--;

line 17, change "(B)" to --39(B)--;

line 19, change "(B)" to --40(B)--;

line 22, change "(B)" to --41(B)--; change "(C)" --41(C)--.

Page 58, line 4, change "(B)" to --43(B)--;

line 6, change "(B)" to --44(B)--; change "(C)" to --44(C)--;

line 11, change "(B)" to --46(B)--;

line 14, change "(B)" to --47(B)--;

line 20, change "(B)" to --49(B)--; change "(C)" to --49(C)--;

line 24, change "(B)" to --51(B)--; change "(C)" to --51(C)--.

Page 59, line 4, change "(B)" to --52(B)--;

line 22, change "(B)" to --63(B)--; change "(C)" to --63(C)--.

Page 71, line 3, change "seqq." to --seq.--;

line 6, change "(cf. Fig. 1) to --(Fig. 1)--;

line 21, change "which constitutes" to --in--;

line 24, change "(here typically" to --(for example--; change "(here" to --(for--.

Page 72, line 1, change "typically" to --example--;

line 3, change "which constitutes" to --in--;

line 4, change "(here typically" to --(for example--;

line 6, change "self" to --"self"-- ; change "(here typically" to --(for example--;

line 7, change "(here typically" to --(for example--;

line 8, change "(here typically" to --(for example--;

line 20, change "which constitutes" to --in--;

line 22, delete "typically".

Page 73, line 1, change "which" to --in--;

line 2, delete "constitutes";

line 7, change "which constitutes" to --in--;

line 8, delete "typically";

line 11, change "which constitutes" to --in--.

Page 74, line 15, after "of" insert--the--;

line 18, after "element" insert --ME$_B$ in Fig. 3--; delete "in its column";

line 19, delete "of";

line 20, change "indicative of" to --indicating--.

Page 75, line 6, after "are" insert --the--;

line 7, delete "one".

Page 76, line 2, delete "a";

line 3, change "the" to --an-- (both occurrences);

line 14, delete "of" (first occurrence); after "of" insert --the--.

Page 78, line 6, change "is" to --may be--.

Page 80, line 20, change "elements" to --elements,--.

Page 81, line 1, after "of" (second occurrence) insert --the--;

line 4, change "why it is to do so is" to --this is done is so--.

Page 83, line 2, change "why it is to do so is" to --this is done is so--; change "including" to --in Fig. 11 and--;

line 3, change "are" to --can be--.

Page 189, line 19, change "th" to --the--.

IN THE CLAIMS:

Please CANCEL original claims 1, 2, 11, 12, 34, and 39-57 without prejudice or disclaimer and AMEND the claims as follows.

3. (ONCE AMENDED) An object-oriented programming apparatus for interconnecting a plurality of objects each having data and operations, said object-oriented programming apparatus comprising:

instruction coupling means for permitting a transfer of messages between a first object, having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to the messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of providing such] detecting a correspondence [that] between the message of the first object [is associated with] the method of the second object; and

[an] input instruction tag table generating means for generating an input instruction tag table indicating an association of messages of [another] the at least one other object with methods of [self] the first object upon detection of the correspondence therebetween by said instruction coupling means, for each other object generating one of the messages, on the output instruction bus portion of [self] the first object.

4. (ONCE AMENDED) An object-oriented programming apparatus according to claim 3,

wherein said instruction coupling means generates a method element list in which arranged are method elements including a method ID for specifying [a] the method of [another] the second object associated with [a] the message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said input instruction tag table generating means generates the input instruction tag table and adds the input instruction tag table to the method elements including the pointer to [another] the second object associated with the input instruction tag table.

5. (ONCE AMENDED) An object-oriented programming apparatus for interconnecting a plurality of objects each having data and operations, said object-oriented programming apparatus comprising:

instruction coupling means for permitting [a] transfer of messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of providing such] detecting a correspondence [that] between the message of the first object [is associated with] and the method of the second object; and

an output instruction tag table generating means for generating an output instruction tag table indicating an association of methods of [another] the at least one other object with messages of [self] the first object upon detection of the correspondence therebetween by said instruction coupling means, for each other object receiving one of the messages, on the output instruction bus portion of [self] the first object.

6. (ONCE AMENDED) An object-oriented programming apparatus according to claim 5, wherein said instruction coupling means generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the

second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said output instruction tag table generating means generates the output instruction tag table and adds the output instruction tag table to the method elements including the pointer to [another] the second object associated with the output instruction tag table.

7. (ONCE AMENDED) An object-oriented programming apparatus for interconnecting a plurality of objects each having data and operations, said object-oriented programming apparatus comprising:

instruction coupling means for permitting [a] transfer of messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of providing such] detecting a correspondence [that] between the message of the first object [is associated with] and the method of the second object; and

an input data tag table generating means for generating, upon detection of the correspondence between the message of the first object and the method of the second object by said instruction coupling means, an input data tag table indicating an association of a data element list ID for identifying a data element list in which pointers to data storage areas for storing data are arranged with a pointer element list ID for identifying a pointer element list in which pointers to data storage areas for storing [pointer] pointers to data are arranged, for each other object, on the output instruction bus portion of [self] the first object.

8. (ONCE AMENDED) An object-oriented programming apparatus according to claim 7, wherein said instruction coupling means generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the

second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said input data tag table generating means generates the input data tag table and adds the input data tag table to the method elements including the pointer to [another] the second object associated with the input data tag table.

9. (ONCE AMENDED) An object-oriented programming apparatus for interconnecting a plurality of objects each having data and operations, said object-oriented programming apparatus comprising:

instruction coupling means for permitting [a] transfer of messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of providing such] detecting a correspondence [that] between the message of the first object [is associated with] and the method of the second object; and

an output data tag table generating means for generating, upon detection of the correspondence between the message of the first object and the method of the second object by said instruction coupling means, an output data tag table indicating an association of a pointer element list ID for identifying a pointer element list in which pointers to pointer storage areas for storing pointers to data are arranged with a data element list ID for identifying a data element list in which pointers to data storage areas for storing data are arranged, for each other object, on the output instruction bus portion of [self] the first object.

10. (ONCE AMENDED) An object-oriented programming apparatus according to claim 9,

wherein said instruction coupling means generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said output data tag table generating means generates the output data tag table and adds the output data tag table to the method elements including the pointer to [another] the second object associated with the output data tag table.

13. (ONCE AMENDED) An object-oriented program storage medium for storing a plurality of objects each having data and operations, said object-oriented program storage medium storing an object coupling program to control a computer to perform a method comprising: [instruction coupling means for permitting a transfer of]

transferring messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of] providing [such] a correspondence [that] between the message of the first object [is associated with] and the method of the second object; and [an input instruction tag table generating means for]

generating an input instruction tag table indicating an association of messages of [another] the at least one other object with messages of [self] the first object, for each other object, on the output instruction bus portion of [self] the first object.

14. (ONCE AMENDED) An object-oriented program storage medium according to claim 13,

wherein said [instruction coupling means] transferring generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and wherein said [output instruction tag table] generating [means] generates the output instruction tag table and adds the output instruction tag table to the method elements including the pointer to [another] the second object associated with the output instruction tag table.

15. (ONCE AMENDED) An object-oriented program storage medium according to claim 14, wherein the first object having [the] a method element to which the input instruction tag table is added calls, when calling the second object identified by the method element, the second object giving as arguments the method ID and the input instruction tag table which are stored in the method element.

17. (ONCE AMENDED) An object-oriented program storage medium according to claim 15, wherein the second object receives the messages directed from the first object to the second object, and refers to the input instruction tag table, which is an argument of the received message, to add the method element related to the method of the first object associated with the message of the second object to the method element list of the second object associated with the message of the second object.

18. (ONCE AMENDED) An object-oriented program storage medium according to claim 15, wherein the second object [has means for producing] produces a third object, receives the messages directed from the first object to the second object, and refers to the input instruction tag table, which is an argument of the received message, to add the method element related to the method of the first object associated with messages of the third object to the method element list of the third object associated with the message of the third object.

19. (ONCE AMENDED) An object-oriented program storage medium for storing a plurality of objects each having data and operations[, said object-oriented program storage medium storing] and an object coupling program to control a computer to perform a method comprising: [instruction coupling means for permitting a transfer of]

transferring messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of] providing [such] a correspondence that between the message of the first object [is associated with] and the method of the second object; and [an output instruction tag table generating means for]

generating an output instruction tag table indicating an association of methods of another object with messages of self object, for each other object, on the output instruction bus portion of [self] the first object.

20. (ONCE AMENDED) An object-oriented program storage medium according to claim 19,

wherein said [instruction coupling means] transferring generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said [output data tag table] generating [means] generates the output instruction tag table and adds the output instruction tag table to the method elements including the pointer to [another] the second object associated with the output instruction tag table.

21. (ONCE AMENDED) An object-oriented program storage medium according to claim 20, wherein the first object having [the] a method element to which the output instruction tag table is added calls, when calling the second object identified by the method element, the second object giving as arguments the method ID and the output instruction tag table which are stored in the method element.

23. (ONCE AMENDED) An object-oriented program storage medium according to claim 21, wherein the second object [has means for producing] produces a third object, receives messages directed from the first object to the second object, and refers to the output instruction tag table, which is an argument of the received message, to add the method element related to the method of the third object associated with messages of the first object to the method element list of the first object associated with the message of the first object.

24. (ONCE AMENDED) An object-oriented program storage medium for storing a plurality of objects each having data and operations[, said object-oriented program storage medium storing] and an object coupling program to control a computer to perform a method comprising: [instruction coupling means for permitting a transfer of]

transferring messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of] providing [such] a correspondence that between the message of the first object [is associated with] and the method of the second object; and [an input data tag table generating means for]

generating an input data tag table indicating an association of a data element list ID for identifying a data element list in which pointers to data storage areas for storing data are arranged with a pointer element list ID for identifying a pointer element list in which pointers to data storage areas for storing [pointer] pointers to data are arranged, for each other object, on the output instruction bus portion of [self] the first object.

25. (ONCE AMENDED) An object-oriented program storage medium according to claim 24,

wherein said [instruction coupling means] transferring generates a method element list in which arranged are method elements including a method ID for specifying a method of [another] the second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said [input data tag table] generating [means] generates the input data tag table and adds the input data tag table to the method elements including the pointer to [another] the second object associated with the input data tag table.

26. (ONCE AMENDED) An object-oriented program storage medium according to claim 25, wherein the first object having [the] a method element to which the input data tag table is added calls, when calling the second object identified by the method element, the second object giving as arguments the method ID and the input data tag table which are stored in the method element.

27. (ONCE AMENDED) An object-oriented program storage medium according to claim 26, wherein the second object receives messages directed from the first object to the second object, refers to the input data tag-table, which is an argument of the received message, to obtain the pointer element list ID of the first object, produces the pointer element list identified by the pointer element list ID[,] of the first object and [in addition] the data element list identified by the data element list ID associated with the pointer element list ID[,] of the second object, and writes the pointers arranged in the data element list of the second object into the pointer storage areas indicated by the pointers arranged in the pointer element list of the first object.

28. (ONCE AMENDED) An object-oriented program storage medium according to claim 26, wherein the second object [has means for producing] produces a third object, receives messages directed from the first object to the second object, refers to the input data tag table, which is an argument of the received message, to obtain the pointer element list ID of the first object, produces the pointer element list identified by the pointer element list ID, of the first object and [in addition] the data element list identified by the data element list ID

associated with the pointer element list ID[,] of the third object, and writes the pointers arranged in the data element list of the third object into the pointer storage areas indicated by the pointers arranged in the pointer element list of the first object.

29. (ONCE AMENDED) An object-oriented program storage medium for storing a plurality of objects each having data and operations[, said object-oriented program storage medium storing] and an object coupling program to control a computer to perform a method comprising: [instruction coupling means for permitting a transfer of]

transferring messages between a first object having an output instruction bus portion for performing [a] processing for [an] issue of messages directed to [another] at least one other object and a second object having an input instruction bus portion responsive to messages issued by [another] the first object and directed to [self] the second object for activating a method of [self] the second object associated with [the] each received message, by [means of] providing [such] a correspondence that between the message of the first object [is associated with] and the method of the second object; and [an output instruction tag table generating means for]

generating, upon detection of the correspondence between the message of the first object and the method of the second object by said instruction coupling means, an output data tag table indicating an association of a pointer element list ID for identifying a pointer element list in which pointers to pointer storage areas for storing pointers to data are arranged with a data element list ID for identifying a data element list in which pointers to data storage areas for storing data are arranged, for each other object, on the output instruction bus portion of [self] the first object.

30. (ONCE AMENDED) An object-oriented program storage medium according to claim 29, [wherein said instruction coupling means generates]

further comprising generating a method element list in which arranged are method elements including a method ID for specifying a method of [another] the second object associated with a message of [self] the first object, and a pointer to [another] the second object in which the method specified by the method ID is executed, and

wherein said [output data tag table] generating [means] generates the output data tag table and adds the output data tag table to the method elements including the pointer to [another] the second object associated with the output data tag table.

32. (ONCE AMENDED) An object-oriented program storage medium according to claim 31, wherein the second object receives messages directed from the first object to the second object, refers to the output data tag table, which is an argument of the received message, to obtain the data element list ID of the first object, produces the data element list identified by the data element list ID, of the first object and [in addition] the pointer element list identified by the pointer element list ID associated with the data element list ID, of the second, and writes the pointers arranged in the data element list of the first object into the pointer storage areas indicated by the pointers arranged in the pointer element list of the second object.

33. (ONCE AMENDED) An object-oriented program storage medium according to claim 31, wherein the second object [has means for producing] produces a third object, receives messages directed from the first object to the second object, refers to the output data tag table, which is an argument of the received message, to obtain the data element list ID of the first object, produces the data element list identified by the data element list ID[,] of the first object and in addition the pointer element list identified by the pointer element list ID associated with the data element list ID[,] of the third object, and writes the pointers arranged in the data element list of the first object into the pointer storage areas indicated by the pointers arranged in the pointer element list of the third object.

Please ADD the following new claim:

58. (NEW) A method performed by a computer to establish communication between first and second objects, each having data and at least one method, comprising:

detecting a correspondence between a message of the first object and a method of the second object;

generating a data element list for first pointers to data storage areas in the first object;

generating a pointer element list for second pointers to pointer storage areas in the second object;

generating a data tag table associating a data list identifier with a pointer list identifier, the data list identifier identifying the data element list for the first object, and the pointer list identifier identifying the pointer element list for the second object; and

after the first pointers have been stored in the data element list, using the data list identifier and the pointer list identifier to write the first pointers into the pointer storage areas indicated by the second pointers for reference by the second object to obtain the data in the first object.

REMARKS

This Preliminary Amendment is submitted to improve the form of the specification as originally filed. No new matter is presented herein.

It is respectfully requested that this Preliminary Amendment be entered in the above-referenced application.

If any further fees are required in connection with the filing of this Preliminary Amendment, please charge same to our Deposit Account No. 19-3935.

Respectfully submitted,

STAAS & HALSEY

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

Dated: 1/22/01

700 Eleventh Street, N.W.
Washington, D.C. 20001
Telephone: (202) 434-1500